Contextual Information Retrieval using Concept Chain Graphs

Rohini K. Srihari, Sudarshan Lamkhede, Anmol Bhasin, and Wei Dai

State University of New York at Buffalo, Buffalo, NY 14228, USA,
rohini@cedar.buffalo.edu

Abstract. This paper discusses concept chain graphs, a new framework for information retrieval that supports sophisticated contextual queries. Concept chain graphs subsume traditional information retrieval models, but extend them by supporting (i) more sophisticated content representation reflecting information extraction output, and (ii) more sophisticated retrieval algorithms including probabilistic graph models and graph mining. Concept chain graphs are designed specifically for applications involving unapparent information revelation (UIR). UIR manifests itself when information generated by multiple authors working independently at different times may together reveal more information than apparent. A key to UIR is connecting information trails that span multiple documents. This requires the support of sophisticated models of context, including cross-document context. Three types of queries are discussed in this paper: (i) concept-based queries using ontologies, (ii) concept chain queries which find the best evidence trail connecting two concepts across documents, and (iii) concept graph queries which reflect more complex patterns. Examples from processing the 9-11 corpus are discussed.

1 Introduction

In order to facilitate the use of more context in information retrieval, it is necessary to develop richer IR models. Richness comes both in the form of sophisticated content representation, i,e. going beyond the traditional bag-of-words model, as well as more sophisticated retrieval algorithms that support context-based queries. In this paper, we are focused on various types of contextual queries, but all of them share the property that no single document can suffice as the response to a query. We refer to this special case of IR/text mining as unapparent information revelation (UIR). Thus, UIR models are focused on returning ranked document sets as opposed to ranked documents.

This paper focuses on UIR problems which are relevant to homeland security. This involves processing a large open source document collection pertaining to the 9/11 attack, including the publicly available 9/11 commission report. The goal is to permit interactive text mining on this corpus leading to intelligence from open sources. It focuses on two types of text mining problems: (i) concept chain detection and (ii) scenario detection. A concept chain query on this corpus looks for best paths connecting, for example, the trucking industry and foreign banks. This should reveal various paths going across multiple documents from say, a truck parts manufacturer through an insurance claim to a foreign bank. A scenario query typically involves patterns of activities that can lead to a security event.

Current techniques for detecting such chains and scenarios are fragile: (i) they rely on an information extraction system to accurately tag key entities and relationships, (ii) they do not take into account more general concepts such as aviation and trucking industry, but are limited to named entities, and (iii) they require users to anticipate and predefine specific scenarios of interest; this in turns involves complex modeling. Analysts cannot anticipate all types of event patterns leading to security events; the system should discover potential patterns! On the other hand, analysts are able to give examples of patterns (occurring in the corpus) involving a set of concepts. The UIR system presented here finds *instances* of the input pattern in the corpus; it then generalizes the matches to produce new patterns of possible interest. The system described here is immediately usable by analysts without having to do cumbersome modeling and customization. We use ontologies that have been developed for the security/counter-terrorism domain. The processing of documents and mapping of extracted concepts into this ontology is automatic.

The UIR problem can be thought of as generalizing the information retrieval (IR) task. In IR systems, the input is a set of keywords, whereas in UIR the query has complex semantics which needs to be interpreted. Two of these queries are described shortly. In IR systems the output is a ranked list of documents, where each document is ranked independently wrt the query. In UIR, the output is ranked sets of documents. In UIR, the documents in a set are ranked both wrt to the query as well as to each other.

We are specifically interested in the following types of UIR queries:

concept chain queries : in this case, the user is specifying the following query which traditional IR systems cannot handle: *find the most plausible relationship between concept A and concept B* assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. Thus, finding the best concept chain is equivalent to finding the best set of documents that connect the concepts. We go one step further and require the response to be a set of text snippets extracted from multiple documents. This is in fact a *cross-document summary* of the plausible relationship between the two concepts.

concept graph queries : this is a generalization of the above, whereby the user may specify a connected graph consisting of several concept chains. Although it would suffice to have just the second type of query, we choose to include both, since different techniques can be used to handle these.

A representation formalism has been developed called concept chain graph (CCG) that has the robustness and scalability of IR frameworks such as the vector space model, in conjunction with richer, probabilistic representation and reasoning frameworks. The solution being pursued takes the view that the concepts and associations in a particular domain can be represented as a probabilistic network with the nodes representing concepts and edges between them representing generic associations; weights can be learned or assigned. A document is viewed as a sub-graph of this probabilistic network.

2 Background

The use of context in IR has traditionally referred to previous search activity by the user. For example, follow-on queries where the satisfaction of a new query must take into account the results of a previous query, or set of queries. Another example is the High Accuracy Retrieval of Documents (HARD), a passage retrieval TREC task where the users can specify the context of a query explicitly. For example a user may be interested only in factual information versus background information. It is also used to model general user interests and preferences which in turn influence the results of a search. More recently, [1] discuss the use of physical context with respect to mobile computing environments. In this paper, we focus on a different interpretation of contextual IR, namely the context provided by concept neighborhoods. We argue that it is necessary to capture and store such context for UIR searches. A key issue is discovering and representing associations between concepts across documents.

There has been work on discovering connections between concepts across documents using social network graphs, where nodes represent documents, and links represent connections (typically URL links) between documents. However much of the work on social network analysis has focused on different types of problems, such as detecting communities [2]. [3] is the work which is closest to the research presented here, at least in its goals. The authors model the problem of detecting associations between people as finding a connection subgraph and present a solution based on electricity analogues. However there are several differences which should be noted. The most notable difference is the reliance on URL links to establish connections between documents. Our approach extracts associations based on content (textual) analysis. Second, the connection subgraph approach presents all paths together, while our approach presents the paths individually. This allows greater user input in determining the *best* paths, including recency, novelty, semantic coherence, etc. Third, the approach presented here attempts to generate an explanation of the chains, whereas the connection subgraph approach does not. Finally, the connection subgraph solution only addresses named entities whereas this approach extends to general concepts. Finally, [4] describes an approach to context sensitive information inference by considering information flows through conceptual space. Although the ideas of modeling context has similar motivations to the work presented here, the focus is still on individual document retrieval.

This leads us to the novelty of the UIR solution presented here. First, it defines a new content representation that is able to: (i) store information about both instances and general concepts such as pilot, and (ii) explicitly store connections between concepts across documents. It is the latter that leads to interesting trails of information. This also entails the need for an information extraction system that can extract both named entities and generic concepts, derive both named and unnamed associations between concepts, as well as map concepts into higher-level ontologies. Thus, even though it is only possible to extract associations between concepts at the document level, one can infer associations between concepts across documents due to the subcategory/supercategory hierarchy. Second, the nature of the representation which captures both statistics about concept/association occurrence frequency as well as the graphical view of information can exploit various methods for finding paths connecting concepts. Thus, very sophisticated probabilistic sequence and graph models for machine learning can be brought to bear, including Markov networks.

3 UIR System

Figure 1 illustrates the overall architecture of the UIR system. It consists of two distinct components, the CCG building component and the UIR toolkit which supports the text mining functionality. It is beyond the scope of this paper to discuss the construction of the CCG in detail. There are several steps to constructing the CCG including: (i) development of a suitable domain ontology, (ii) customizing an information extraction engine such as InfoXtract [5] to extract key concepts and associations, (iii) mapping the concepts into the ontology. InfoXtract tags *named entities* as well as provides subject-verb-object syntactic analysis. The latter is used to select key concepts as well as generate potential associations between concepts. The UIR system also includes a graphical user interface which permits users to interact with the system. The following functionality is currently provided:

– keyword search: documents ranked based on the vector space model
– concept search: concept search permits users to select concepts (rather than keywords) from the domain ontology - the relevant documents are returned
– browse concept neighborhood: users can input concepts - the system displays the neighborhood graph of the concept which includes concepts one or more links away from the target concept. Users can click on a link to see the text snippet that provides evidence for the link.
– concept chain retrieval: users input two concepts and the system generates the documents and evidence trail connecting the two concepts
– concept graph retrieval: users input a connected graph consisting of a set of chains; the system first returns all possible matches in the corpus. The user may select one or more instances to be generalized.

For the experiments we used the 9/11 commission report as the data set. The report consists of Executive Summary, Preface, 13 chapters, Appendix and Notes. Each of them was considered as a seperate document. The Notes were divided into 3 parts so the collection had a total of 19 documents. The entire collection was processed using InfoXtract and concepts were extracted and mapped into a terrorism ontology designed for this purpose. It should be noted that these *documents* are still very long, and hence we encountered difficulties in searching

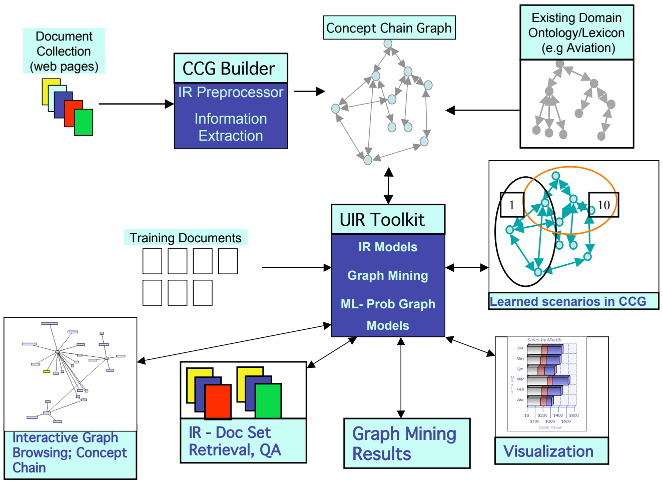

Fig. 1. UIR System Architecture

for good examples where two or more concepts of interest were *not* in the same document. We are now in the process of indexing the document based on logical paragraphs: this also provides a test of scalability of the UIR system.

4 Concept Chain Graphs

Concept Chain Graphs are a new content representation for IR which are conducive to text mining. The Concepts and Associations are represented as a graph; occurrence statistics in a document subset or collection for both concepts and associations is maintained. Different retrieval models and index representations have been developed over the years for Information Retrieval including Vector Space Model (documents and queries are viewed as vectors in an N-dimensional space and query-document similarity is a measure of similarity (e.g. cosine similarity) defined in the vector space), and Probabilistic Models (where documents are ranked based on their relevance to the given query). The vector space model and early probabilistic models have separate indexing mechanisms and retrieval strategies. An Inverted Index is used to store the document occurrence information for each term in the vocabulary. The relations or associations between terms are not represented in this index. This effort is focused on an integrated indexing/retrieval model such as that used in the INQUERY[6] system (based on Bayesian Networks). However Bayesian networks are limited to directed graphs;

our effort considers both directed and undirected associations. Other desired features of the CCG include: (i) the ability to support both IR and UIR queries, (ii) the ability to exploit graph topology as well as statistical information, (iii) support for various probabilistic models (e.g Hidden Markov Models, Markov Random Fields) and machine learning, (iv) scalability and robustness similar to IR models, (v) support for incorporating domain ontologies, and (vi) support for interactive browsing and querying.

Formally a CCG is a hypergraph $G(E, V)$ with E edges and V nodes representing a set of documents D with the following properties:

- each node v represents a term, a concept or a document
- each edge e represents an association between two concepts or a membership link (e.g. link between a document and a concept or links between a concept and its contituent terms)

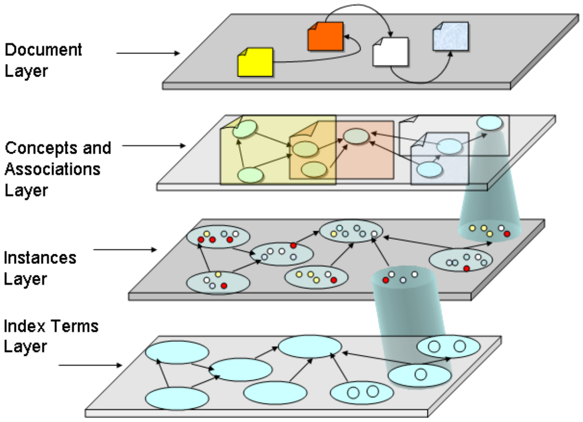

Fig. 2. Concept Chain Graph Layers

Concept Chain Graph is a hierarchical combination of index terms, concepts, associations and documents using directed as well as undirected links and can be viewed as consisting of four layers as shown in figure 2:

1. Document Layer. This layer contains documents and links(e.g. hyperlinks) between them.
2. Concepts and Associations Layer. Consists of concepts and associations coming from corpus or ontology mapping.
3. Instances Layer. Tracks instances of concepts and associations detected in the corpus back to documents. Also maintains instance specific information last offsets and type.

4. Index Terms Layer. Consists of index terms and hits.

Since the CCG combines both topological information and statistics, it can be used to derive various models of *context*, and hence context similarity. For example, the context of a concept can include surrounding concepts (say those one link away), and/or statistical co-occurence information obtained from the corpus. We have successfully used the CCG framework to implement both an IR model, namely the Vector Space Model for ranking documents, as well as a UIR model for satisfying concept-chain and concept-graph queries.

CCG infrastructure consists of two main components. The first component is responsible for concept extraction and selection, construction of the CCG and other models defined on it, various search and retrieval functions that let other applications accessing the CCG (such as the concept chain query module) view the CCG as needed. It also incorporates a command line user interface and is fully customizable. This component is written in ISO C++ and compiled using GCC. The current implementation utilizes a package, E4Graph[1] which was chosen principally for its abilities to store graph-like data persistently and to access and manipulate that data efficiently. Features of E4Graph can be exploited to create hypergraphs which are key in abstracting the different layers of the index.

The second component, which is written in Java, consists of the GUI and query preocessing module for Concept Graph Queries. It uses SUBDUE[7] as an external component for graph matching. It is also responsible for providing vaious views of underlying CCG and communicates with the first module using JNI. Currently both components are hosted on a RedHat Linux 9.0 operating system, running on i386 platform.

5 UIR Queries

Based on the CCG, it is possible to process various types of UIR queries. We use a combination of vector space models, probabilistic graph models, and graph mining in satisfying these queries. The three specific UIR queries are discussed in more detail.

5.1 Concept Queries

Concept queries are an enhancement of keyword-based queries but use the concept ontology in expanding the set of terms that are relevant. While query expansion techniques have been widely investigated, the expansion here is through a very focused ontology. Only concepts which correspond to the general terms are used in expansion. An example is a search for *wmd*. This returns documents that do not contain the term wmd but do contain terms such as *biological weapons* as shown below.

[1] http://e4graph.sourceforge.net/

```
Doc 1: Chapter 12 Global Stratergy
Pakistan posseses nuclear weapons and frighteningly..
Al-Qaeda had tried to make or acquire nuclear weapons..

Doc 2: Notes
Bin Ladin links to material related to WMD..
        Sufat did not start the al_qaeda Biological Weapons program until..

Doc3 : Chapter 4 Responses to initial assaults
Perhaps using Weapons of Mass Destruction..
Some intelligence reports mentioned chemical weapons..
```

The UIR toolkit permits more context to be visualized surrounding a concept as illustrated by figure 3. It shows all concepts which are at distance 1 from WMD. The system permits users to click on any arcs in order to show the snippet of text that provides evidence for the association. Users can also increase the number of links resulting in larger neighborhoods.

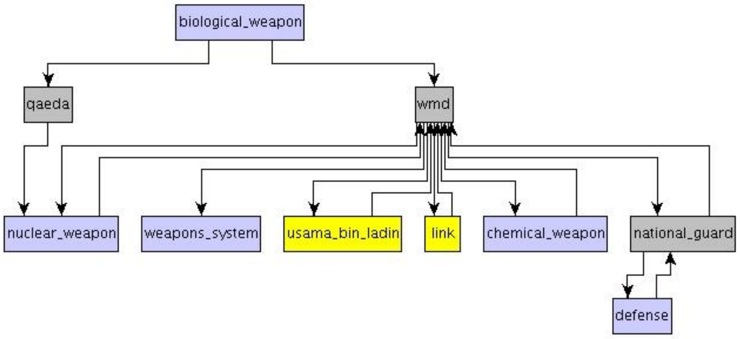

Fig. 3. Concept Neighborhood for WMD

5.2 Concept Chain Queries

We use probabilistic graph models to find the best concept chains. It involves a 2-level Markov model defined on the CCG: the first level uses a Markov Chain representation in order to find the best chain of a given length connecting two concepts. It uses a second, HMM in order to retrieve the best set of documents that could have generated the chain. Each of these levels is now described in more detail.

We use a Markov Graph of Concepts and Association which we refer to as a Concept Markov Graph (CMG). All the concept nodes in the CCG are treated as the random variables of a CMG and all the associations are used as transitions among them. A concept can make a transition to another concept through one of the "recognized relations". If a concept X is related to another concept Y which has a similar context as that of X, then such a relation can be coherent and meaningful. Each link can be seen as some *drift* away from the original concept. Keeping this in mind we calculate the transition probabilities of the concepts based on their contexts. We define the context of a concept formally as:

A concept can be formed by one or more terms. These terms define a term vector for that concept in the n dimensional Vector Space Model. A context of a concept is given by the union of term vectors of the concept and the term vectors of its related concepts.

Now for any concept C_i and C_j, the transition probability is given as

$$P\left(C_i, C_j\right) = \frac{sim\left(\overline{C_i}, \overline{C_j}\right)}{\sum sim\left(\overline{C_i}, \overline{C_k}\right)} \quad \forall C_k \in \{neighbors\left(C_i\right)\} \tag{1}$$

Where $sim\left(\overline{C_i}, \overline{C_j}\right)$ is the similarity between the context vectors of concept C_i and concept C_j. It is important to note that even though the similarity measures are symmetric in that $sim(\overline{C_i}, \overline{C_j}) = sim(\overline{C_j}, \overline{C_i})$, the transition probabilities are not symmetric i.e. $P(C_i, C_j) \neq P(C_j, C_i)$. This asymmetricity arises from the fact that each concept has a different neighborhood. It is in a way interesting to have the forward probabilities differ from the backward probabilities in that it gives a possibility to get a different best Markov Sequence from C_i to C_j than from C_j to C_i.

The user specified concepts are passed on to the CMG as starting state and absorbing state. The best chain from the source to destination is computed as the best Markov sequence in CMG using the Viterbi algorithm. The next step is to collect evidence that supports the concept chain. This evidence is simply a set of documents in which the concept chain occurs. Our goal is to provide users with a ranked list of document sets. For this purpose, a Hidden Markov Model (HMM) is defined on the CCG. We call this Document Set Retrieval Hidden Markov Model (DocSetHMM).

Hyperlinks give users the freedom to jump to any document but at the same time they restrict the number of transitions from a document. Whereas in a plain text corpus, we have to keep in mind that users may want to read any document from the corpus after the current document. Thus, every document must allow for a transition to every other document in the corpus. It is quite possible that a user may want to stay in the same document or go to a different section of the same document. Hence self transitions must be allowed. Keeping these issues in mind, the document transition probabilities are calculated. Let D_i and D_j be any two documents in the collection, the transition probability between them is given by:

$$P\left(D_i, D_j\right) = \frac{sim\left(D_i, D_J\right)}{\sum_{k=1}^{n} sim\left(D_i, D_k\right)} \tag{2}$$

where, n is the total number of documents in the corpus. Self transition probability is $P(D_i, D_i) = 1.0$

Emission probability in DocSetHMM is the probability of observing a concept or an association given a document. For an association A_i, the emission probability from document D_j is

$$P\left(A_i|D_j\right) = \frac{count\left(A_i, D_j\right)}{\sum_{A_k} count\left(A_k, D_j\right)} \tag{3}$$

where, $count\left(A_i, D_j\right)$ gives the total number of times A_i occurs in D_j i.e. the count of instances of A_i in D_j. The summation in the denominator gives the total number of association instances in the document.

We experimented with several concept chain queries, including those connecting two named entities, those connecting a named entity to a general concept, and those that connected two general concepts. An example of the last category is presented here. The chain of 4 associations found for ***cockpit_voice_recoder*** and ***hijacking*** is
cockpit_voice_recorder — ***indication*** — ***communication*** — ***movie*** — ***hijackings***

The explanation of this lies in the following excerpts from the report coming from multiple documents:

"... No evidence of firearms or of their identifiable remains was found at the aircraft's crash site, and the cockpit voice recorder gives no indication of a gun being fired or mentioned at any time. ..."

"... Khallad adds that the training involved using flight simulator computer games, viewing movies that featured hijackings, and reading flight schedules to determine which flights would be in the air at the same time in different parts of the world. ..."

The commonality between the concepts *cockpit_voice_recorder* and *hijacking* goes through the concepts *movies* and *communication.* This chain is a good example of use of ontological relations to infer links between two seemingly unconnected concepts.

5.3 Concept Graph Queries

Concept Graph queries are processed by considering every connected pair of vertices and retrieving a set of chains of lengths ranging from 2 to 6 between them. A set of Concept Graphs is then constructed by using a combination of these chains.These Graphs model the query in terms of the associations in the CCG. Interestingly, these set of graphs essentially model all possible scenarios that connect the query concepts by providing instantiations in the corpus of patterns of activity and it does not involve laborious modeling by the user, merely some domain knowledge. This has extremely high relevance to intelligence analysis work.

The system goes one step further and mines similar graph toplogy exploiting the graphical nature of the CCG. Upon selection of one of the model graphs of interest by the user, the system records the graph topology as well as the Concept Types in the structure. An external system SUBDUE[7] is utilised for generation of Isomorphic subgraphs to the query Concept Graphs. The isomorphism is determined based on Concept type matches and the association linkages between them. The Isomorphic subgraph in turn provides a similar scenario to the query scenario. Currently, only node similarity is being considered in generalizing the patterns. We are experimenting with arc similarity also. Since each association can be represented by a set of sentences from the corpus, it is possible to use statistical contextual similarity techniques like LSA in determining whether two edges are sufficiently similar. Finally, users can provide an overlap option and an approximation measure for the Subgraph Isomorphism process to control the degree of similarity.

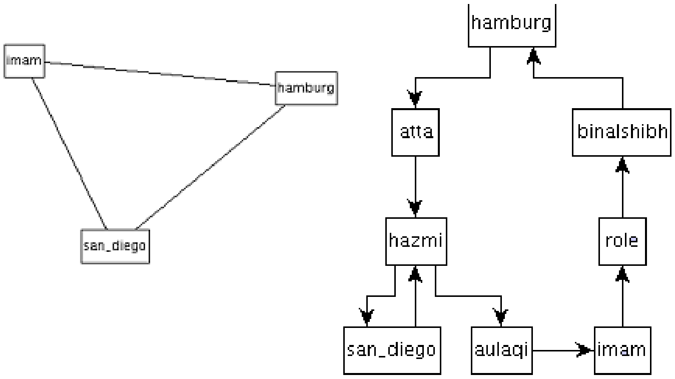

Fig. 4. Concept Graph Query (A)

The Concept chain graph experiments use a connected graph of concepts of interest and perform both: (i) model generation (instantiation) as well as (ii) model matching (generalization). The model generation example in Figure 4 shows the user input on the left side and the model generated on the basis of the CCG on the right. The user query tries to model a relationship between Hamburg, San Diego and Imam (A man leading prayers in the mosque).The model generated by the system on the basis of the 911 corpus can be best described as: (i) BinalShibh and Atta shared apartments in Hamburg,Germany, (ii) Atta and Hazmi were hijackers involved in the 9/11 attacks, and (iii) Hazmi found an apartment in San Diego with a help of an Imam by the name of Anwar Aulaqui at Rabat mosque in San Diego. We do not include examples of the second stage here for space reasons. The system is able to generalize the model to other instances, based on subgraph isomorphism involving concept type matches. Currently, this generates several graphs, where named entity instances are replaced by other instances. We are in the process of implementing the arc similarity match and feel this will provide more interesting generalizations.

6 Conclusions

This paper has presented a system for unapparent information revelation and shown several UIR queries, which are unique since they cannot be satisfied by a single document. Concept chain graphs, which explicitly store context surrounding concepts have been introduced as the basis for advanced contextual queries. This context reflects multiple documents. Examples from the 9-11 corpus have been presented.

Future work includes more quantitative benchmarking of the UIR system by applying it to standard data sets such as the Document Understanding Conference (DUC) for cross-document summarization. We are also working on enhancing the solutions to UIR queries by incorporating more powerful probabilistic graph models such as Markov Random Fields. The goal is to use this framework for machine learning where the system can discover patterns of activity that may warrant attention, as well as quantify information.

7 Acknowledgments

This work is sponsored by NSF grant IIS-0325404 and a research grant from the FAA 032-G-009. The authors would like to thank Nick Schwartzmeyer for his assistance in creating ontologies.

References

1. G.J.F.Jones, P.J.Brown: Information access for context-aware appliances. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000). (2000) 382–384
2. Gibson, D., Kleinberg, J., Raghavan, P.: Inferring web communities from link topology. In: Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia. (1998) 225–234
3. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In et al, K., ed.: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD-2004, Seattle, US, ACM Press, New York, US (2004) 118–127
4. Song, D., Bruza, P.: Towards context-sensitive information inference. Journal of the American Society for Information Science and Technology **54** (2003) 321–334
5. Srihari, R.K., Li, W., Niu, C., Cornell, T.: Infoxtract: A customizable intermediate level information extraction engine. In: Proceedings of the NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS), Edmonton, Canada (2003)
6. Callan, J.P., Croft, W.B., Harding, S.M.: The INQUERY retrieval system. In: Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications. (1992) 78–83
7. Cook, D.J., Holder, L.B.: Graph-based data mining. IEEE Intelligent Systems **15** (2000) 32–41